RECONSTRUCTING THE FUTURE: CAPACITY PLANNING WITH DATA THAT'S GONE TROPPO*

Steve Jenkin
Info Tech,
PO Box 48, Kippax ACT 2615
sjenkin@canb.auug.org.au

Neil J. Gunther
Performance Dynamics Company,
Castro Valley, CA 94552, USA
www.perfdynamics.com

Two of our coworkers, Larry and Katrina, were recently fired for being discovered in a compromising situation at the office. In an act of revenge before leaving, they deleted several directories—some of which contained our validated forecasting data and capacity planning models. Fortunately, our company had the foresight to implement disaster recovery data centers: one in Innisfail, Far North Queensland and the other in New Orleans. We've since joined gamblers anonymous. This still left us with a significant problem. What do you do with capacity planning models that worked in the past, but now some of the historical data is no longer accessible? This paper presents techniques for reconstructing your capacity planning models when significant chunks of data are missing. In our case, we were even able to improve our forecasting with a more accurate power-law model, constructed by virtue of gaining deeper insight during the reconstruction phase.

1 INTRODUCTION

This paper presents a case study demonstrating how we reconstructed capacity planning models when significant chunks of the original data used to develop those models disappeared because things in the environment had unexpectedly "gone troppo" (see the Glossary of Terms in Appendix B). Obviously, if everything has been completely obliterated, nothing can be done. More commonly, however, the situation resembles that created by topical storms like "Larry" and "Katrina"; massive devastation in the beginning with asymptotic recovery in the end, depending on the cost and time invested.

As long as some of the original historical data used to validate the original capacity planning models remains accessible, all is not lost. Data loss can occur for many unexpected reasons. You have vindictive colleagues like ours, you lost your job, you have a new job but this company doesn't use the same tools as your previous employer, you fat-fingered `rm -rf *`, you can't remember where you put the data, or it's encrypted and you forgot the password? Our methods are based on nonlinear and multivariate techniques. An entirely unforeseen consequence of our reconstruction effort was that we uncovered both some surprises and new understandings of the application. For example, important performance targets that had been overlooked during the original project now became self-evident. These targets will have an important bearing on our future capacity planning.

This paper is organized as follows. The essential case study background is presented in Section 2, with the details relegated to Appendix A. Analysis of the remaining raw data is discussed in Section 3. Reconstruction of the original capacity planning models is discussed in Section 3.1 along with some suggestions for better data preservation. Section 4 describes how some of the reconstruction techniques might be further developed and Section 5 presents some general conclusions. A glossary of terms is provided in Appendix B.

2 BEFORE THE STORM

The case study data is taken from the web site hosting the initial Australian GST (Goods and Services Tax) registrations in year 2000. Businesses were required to register with the Australian Tax Office (ATO) for an ABN (Australian Business Number) to claim rebates of input taxes paid. There were five parties involved in the project:

the ATO, the software contractor, the Secure Hosting Facility, BEP (Business Entry Point) [1] and the operations and administration staff.

The collected data contained 480,000 full registrations. On-line registrations started being accepted around September 1999. Of the 3.3 million initial registrations, about 600,000 were processed via the Web, making them the first large-scale set of on-line "transactions" handled by the Federal Government. In the months following the introduction of GST, the same web-site processed 50% of all ABN registrations. More detailed background information on this aspect is provided in Appendix A.

2.1 The Big Picture

The data falls into two periods:

Relic Data: The surviving data available to us for reconstructive analysis. The period covers March 27 to September 19, 2000 (Fig. 1).

Pre-Troppo: Data prior to March 27, 2000 which was used to generate our original capacity planning models and is now missing.

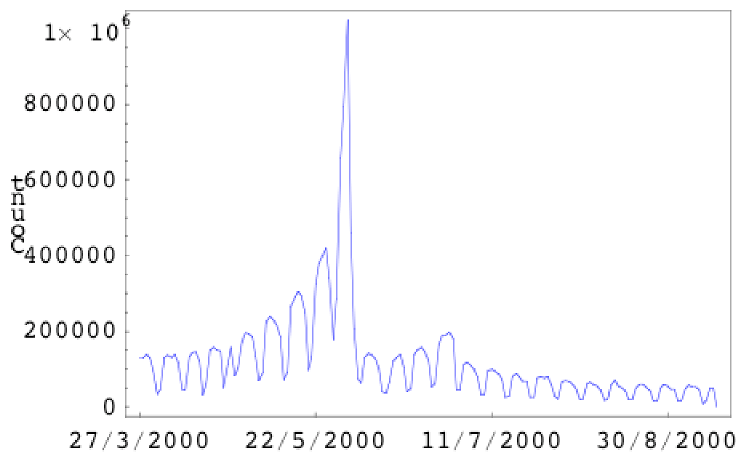


Figure 1: Relic time series of database connections.

We recall several distinct phases that were contained in the pre-troppo data:

Early: August to mid-December, 1999

Xmas Break: Mid-December 1999 to end of January, 2000

Pre-Advertising: 1 February to 27 February (Sun), 2000

Site Overload: 28 (Mon) and 29 (Tue) February, 2000 but not a spike

Those missing data were used to construct a validated exponential growth model (see Sect. 3.2) based on the success of similar techniques described in [Gun01]. The motivation was to introduce a capacity planning model into the ATO project and thereby demonstrate its usefulness to software architects and engineers. Indeed, this was the only sane aspect of the new ATO registration process which did not draw undue attention when things fell apart.

The missing data included logs for the web server and the database server, as well as detailed system performance metrics such as those reported by *vmstat* and *iostat*. Therefore, we knew we could not recreate a detailed system model based on such things as the instantaneous load average, CPU busy, memory consumption and disk utilization, etc. We did, however, have enough relic data to reconstruct reasonably accurate models of total system load expressed as the number of database connections. This enabled us to recover compound growth, the effects of

[1] The Office of Small Business, then part of the Dept. of Employment, Workplace Relations and Small Business, ran a secure web hosting facility for more than thirty other government instrumentalities. All other sites were under active development and maintenance during the GST registration period.

holidays, the normal daily/weekly traffic patterns and, most importantly, the *Christmas Rush* effect of deadlines, and possibly "the day after" effect. This provided a picture which is at least self-consistent even though it can never be corroborated by the original traffic forecasts—now obliterated. Moreover, since we are no longer part of the original project, we can't compare our relic-based analysis with current operations.

Other behavioral periods can also be identified in the relic data of Fig. 1:

Post-Advertising: 1 March (Wed) to 30 May (Tue), 2000

Deadline spike: 31 May (Wed), 2000

June 2000: 1 June to 30 June, 2000

Post Deadline: 1 July to end of data at 19 September, 2000

The first step in the recovery process is to try and recall, as accurately as possible, the pertinent parameters and results of the original capacity planning modeling exercise. This is helpful later on when it comes to validating the reconstructed capacity planning model.

2.2 The Way It Was

There were around 20 million data items in the raw connection data and about 1 million in the spike; far too many for any graphing or statistical package. An initial investigation identified that the database connections were individual processes. Hence, a reasonable measure of total system load was available. Although the database logs were lost, the process accounting data provided enough detail to select workload processes and the nature of their activity. In the original capacity planning project, the data was aggregated and summarized using *awk* and *Perl*. Some time was spent analyzing the data from all the web servers. When it became apparent that

- the system clocks weren't accurately synchronized, and
- the systems were on-line at different times and had different processing capability

a simpler solution was sought.

Gnuplot was used to visualize the data and exported in PDF files. An unanticipated problem with simple-minded graphs arose. They are very slow to display. With around 30,000 data points, 5-minute internalized graphs of the more than 100 days of traffic took several minutes to view.

What looked like a pristine data set initially, based on the process that sent completed registrations to the ATO, turned out to be a dead end because sending of emails, which was done on the database server, was scheduled every 6 minutes and limited to 100 at a time. That load was completely disabled to improve system throughput for the most interesting capacity planning period on the busiest day. This scheduling was mandated by the Secure Hosting Facility after a software fault caused a flood of more than 6,000 emails to be issued after a multi-day stoppage. Such an email flood, containing several days of traffic in 10 minutes, brought the facility mail system to its knees with both the outgoing emails and their acknowledgements. The facility also had a considerable number of other users, including several large departments that were severely impacted. The situation was not improved when the software contractor allowed the fault to recur. We chose to treat all other routine loads such as, housekeeping, administration and backups, as background load on the system.

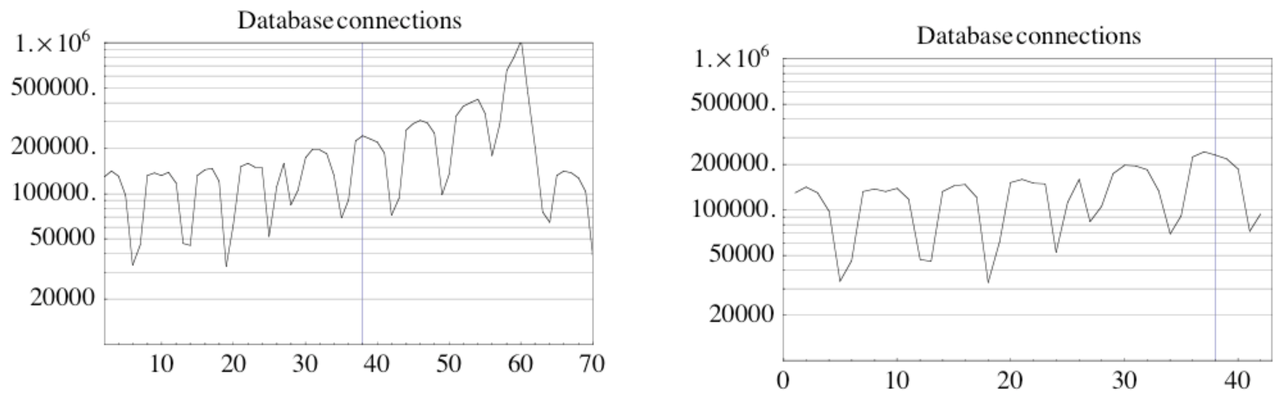
3 AFTER THE STORM

There are three basic steps in conventional data analysis and modelling. (i) Organizing the raw data you plan to model, (ii) perusing your data to determine the best modeling approach to take, and (iii) applying the selected modeling approach and validate it. You also need to state your assumptions, develop a clear idea of what you want to achieve, and be aware of the limitations of the capacity planning models i.e., what they can and cannot produce.

However, very little about our situation is conventional. Our original capacity planning model, along with most of the data used to construct it and validate it, are now missing. Although the above steps can still be largely followed, the emphasis is quite different. We need to pick out from the surviving files (hereafter called *relic* data) those data which can be used to reconstruct the original model as closely as possible. We explain how we did that in Sects. 3.1 and 3.2. An unexpected side-effect of the reconstruction process was the development of new insights and thus an improved capacity planning model in Sect. 3.3.

3.1 Picking Up the Pieces

The first part of our reconstruction effort focused on reproducing the validated exponential growth model alluded to in Sect. 2.1. A fitted exponential curve had been constructed from plotted time series data, the surviving portion of which is shown in Fig. 2 as a semi-log plot. For later analytic convenience, calendar time in Fig. 1 has been replaced by indexed time-steps, which can be read as "days from the start of the relic data window."



(a) Raw data including the observed spike.

(b) Raw data used to predict the spike.

Figure 2: Semi-log plots of the post-troppo subset of historical data in Fig. 1 including the pre-troppo predicted spike at time-step $t = 60$. The portion of the time series that will be used to recreate that prediction now lies in the range $t = 0$ to $t = 38$ (*vertical line*).

The pre-troppo exponential model had already raised concerns about potential overload during the Christmas 1999 period. By late January 2000, however, such overload appeared to have been averted until 29 February when a system meltdown occurred following a round of promotional advertising. That same model also predicted exponential traffic on May 31, 2000; the date when the registration window closed.

One of us (SJ), having noticed the *cascade failure* on 29 February 2000, moved to avoid further embarrassment for the ATO by designing a *Busy Tone* protocol to deter additional incoming users (See Appendix A.2). In fact, as we shall see shortly, the Busy Tone solution was fundamentally flawed and could have promoted a meltdown due to a potential positive feedback cycle. The weakness lay in a decision by the software contractor to use the database to display and control the Busy Tone page. This increased the load by at least 20% during the spike. With increased demand or a minor hardware failure, the system would likely have suffered another melt-down.

3.2 Reconstructing the Exponential Model

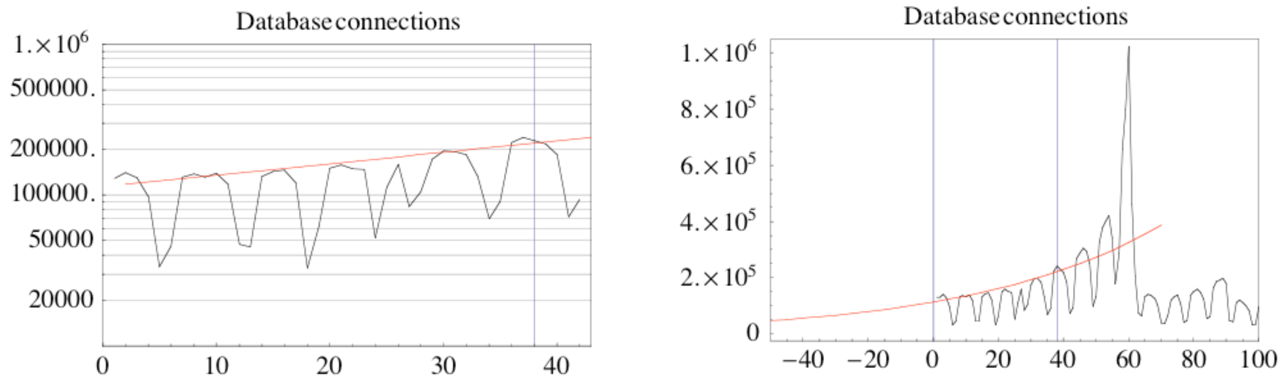
Fig. 2(a) shows a semi-log plot of the relic data preceding the predicted spike on May 31 ($t = 60$ days). It is that growth curve which we attempt to reconstruct in this section. An exponential model:

$$\hat{y}(t) = \alpha e^{\beta t} , \tag{1}$$

where the parameters α and β are fitted to the data, was chosen pre-troppo because it was able to reflect compounded growth in such performance statistics as inbound Internet traffic rates to the web-site, and expected revenue growth [Gun01, Gun06]. Moreover, where a simple linear model might seriously underestimate growth rates, an exponential model can capture the super-linear growth, if it should arise, in terms of the parameter β.

Fig. 2(b) shows the period from March 27 to May 9, 2000 in the relic data. This data, together with the now missing pre-troppo data, was used to parameterize the original exponential model. Recall that a log-plot of a positive exponential function is linear rising and, indeed, we see this characteristic reflected in the trend of the weekly peaks ("humps") in Fig. 2(b). Since we are interested in predicting *peak* load for capacity planning purposes, the exponential model was fitted to the peak number of ORACLE connections for each of the weekly humps up to relic day $t = 38$, as indicated by the cross-hairs in Fig. 3(a).

The exponential model (1) estimates the number of database connections on the y-axis (denoted $\hat{y}$) as an implicit function of time expressed in relic days on the x-axis. It is completely specified by the parameter α, which is merely the height of the curve on the first relic day ($t = 1$) and the parameter β, which corresponds to the growth rate



(a) Exponential model fitted to weekly peaks.

(b) Forward and backward model extrapolations.

Figure 3: Comparison of exponential model (1) with relic data.

(curvature). Mathematica® was used to produce the fitted parameter values:

$$\alpha = 114128 \text{ and } \beta = 0.0175 \,. \tag{2}$$

The growth *doubling period* metric [Gun01] can be determined from the curvature parameter as:

$$T_2 = \frac{Ln(2)}{\beta} = 39.69 \text{ days} \,, \tag{3}$$

or slightly longer than 1 month. Compare this with $T_2 \simeq 6$ months for hypergrowth web-sites prior to the "dot-bomb" bust [Gun06].

There is good news and bad news in this reconstruction. The exponential capacity model is consistent with our pre-troppo data, as indicated in Fig. 3(b) by projecting (1) backwards in time; the pre-troppo period being to the left of zero. It agrees with our recollections in Sect. 3.1, However, the model grossly underestimates the amplitude of the spike in the relic data (to the right of zero). This might be attributable, in part, to the implementation of the Busy Tone protocol mentioned earlier. With the registration window about to close, users expressed their increasing anxiety by continually clicking on the web page until they were allowed entry. Together with the additional load from more database calls implemented by the software contractor, this had the effect of causing the number of database connections to go *super*-exponential! This observation led us to a second, and new, phase in the reconstruction process.

3.3 The Future Ain't What It Used To Be

The appearance of such a sharp spike is reminiscent of so-called *critical phenomena* like that seen in physical systems [GNW80], social systems [Bar03], and even the Internet [Gun05b]. The chief characteristic is that short-range interactions between a few subsystems, otherwise operating independently, somehow become sufficiently correlated to produce long-range order across all subsystems. The onset of such a change or transition in the whole system can occur seemingly spontaneously. Mathematically, this behavior can be expressed in terms of a *power law* model:

$$\hat{y}(t) = \frac{\alpha}{|t - t_c|^{\beta}} \,. \tag{4}$$

If we think of $t_c = 60$ days in Fig. 3(b) as the *critical point* when the spike occurs, it is clear from (4) that the estimated number of database connections, $\hat{y}(t)$, will diverge or surge as $t \to t_c$. This surging is interpreted to mean the system has suddenly gone into a highly correlated state of some sort.

To test this hypothesis, we plotted the relic data (up to t_c) against $|t - t_c|$ on log-log coordinates. The corresponding regression parameters for (4) were determined to be:

$$\alpha = 1.6820 \times 10^6 \text{ and } \beta = 0.6421 \,. \tag{5}$$

Whereas an exponential model becomes linearized on a semi-log plot e.g., Fig. 3(a), a power-law model becomes linearized on a log-log plot. Indeed, Fig. 4 shows rather convincingly that a power law model (the straight line)

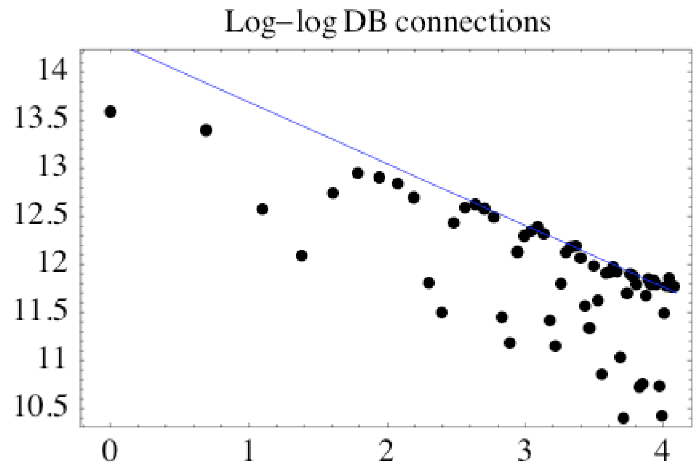
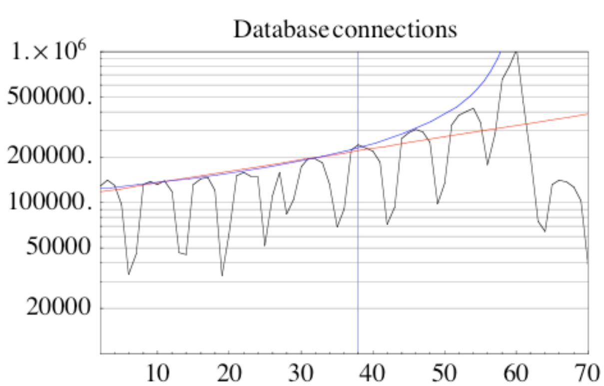


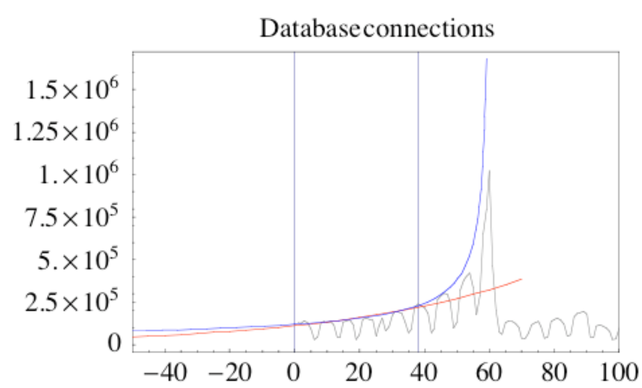
Figure 4: Log-log plot of database connections.

could account for the evolution of *peak* database connections i.e., the upper bound provided by the maximal weekly connections. This approach is also consistent with the way the exponential model was developed. A comparison between the two models in Fig. 5(a) reveals the super-exponential behavior of (4). Fig. 5(b) further indicates that the power-law model could successfully model pre-troppo data. It could also be used to model post-spike data with parameter values slightly different from those in (5), but we don't pursue that aspect here.

It remains to explain how such a highly correlated profile could occur. In Sect. 3.2, it was noted that the number of connections already exceeded the exponential or compounded growth model beyond $t = 40$ days. The people registering with ATO also knew, or were becoming aware, that the registration window was about to close on t_c. It therefore seems quite plausible that we are looking at a *small world* phenomenon where point-to-point interactions in the *social* network e.g., users emailing colleagues or phoning each other to "get registered before it's too late!" actually induced successively more traffic onto the *computer* network. This highly correlated 11th-hour rush of registrations would tend to overload the computer system (including the database server), subsystem queue lengths would increase dramatically, and even users who were already logged on would have more persistent connections because it would take them longer to complete their registrations.



(a) Power law model (*blue*) fitted to peaks in Fig. 2(b).



(b) Forward and backward extrapolations using (4).

Figure 5: Comparison of power law model with relic data.

The upshot of this collective behavior is that it drives the system into a highly correlated state corresponding to the spike around the critical point t_c in the relic data. Here, *highly correlated* means that the system is primarily doing just one thing or persisting in one mode of operation for a longer than average time. Beyond t_c, the registration rush has dissipated and the system relaxes back into a pre-troppo-like profile. The critical behavior associated with the spike is therefore not a physical phase transition i.e., there is no *order parameter* [GNW80]. As mentioned previously, the Busy Tone protocol may also have inadvertently contributed to an additional flurry of login activity.

We also note, in passing, that power law effects are much more highly correlated than the cascade effect mentioned in Sect. 3.1, because the latter is essentially just an exponentiating chain reaction.

4 OTHER DATA RECOVERY TECHNIQUES

What we have presented here is interesting and sufficient for our purposes, but rather modest when compared with the panoply of more sophisticated methods available for recovering and reconstructing data. A complete treatment would take us too far afield, but a brief summary should enable the interested reader to explore some of the alternatives.

The obvious question is, how much data do you need to do a successful reconstruction? Easy to say, not so easy to answer. It depends. In our case, we had one contiguous chunk of relic data to work with, and that might be quite a common situation. But suppose successively more and more uniform strips of data are elided from Fig. 1, for example. How many excised strips would be too many? Suppose further that each strip had a random width. How many irregular excisions would be too many and how would you express that limit?

These questions have a lot in common with methods for reducing the runtime of simulations. In that context the question becomes: Given a random signal, which portions of the signal are essential to include in any simulation? Simulation experts have noted that it is important to try and capture samples of the signal which include *transitions* (both rising and falling), and several heuristics have been developed to accomplish that goal. See for example, the *SimPoint* tool (`www-cse.ucsd.edu/~calder/simpoint/simpoint_overview.htm`). Clearly, SimPoint would identify the critical spike in our data as being a very important transition to include. In a generic time series with less distinct features, however, phase transitions can be very difficult to identify.

Another approach, directed specifically at one-dimensional signal reconstruction, is called *Irregular Sampling*, which is based on Fourier transform and associated wavelet techniques [CRT05, Mal99]. Most of the irregular sampling algorithms are iterative in nature. Starting from some initial sampling-strip guess, further approximations are obtained step by step using the available or assumed knowledge about the size and location of appropriate sampling strips.

For data that is highly periodic or seasonal, the *Holt-Winters* method can be a very effective for data reconstruction [HDA05]. This is essentially a 3-parameter smoothed exponential averaging model [See Gun05a, Chap. 4], and is available in most statistical software packages [Yur98]. See `en.wikipedia.org/wiki/Exponential_smoothing` for a brief but readable introduction to smoothing techniques.

5 CONCLUSION

Things rarely go as planned. During this project we missed some important things. A big surprise in the ORACLE connection data was the additional load imposed by the Busy Tone protocol (See Appendix A.2) along with the unilateral decision by the software contractor to implement it in the database, not as written in the design specification. This was a pernicious fault which luckily was avoided. Just how close we came to another system melt-down on the busiest day due to the mis-implementation of the Busy Tone is chilling in retrospect. In some sense, a lack of system performance instrumentation provided a cushion of blissful ignorance. This approach is not recommended.

The database firewall becoming the rate limiting factor in the system was also a big surprise (See Appendix A.4). Because it was unanticipated, it also was uninstrumented. The other major hardware surprise was the utility we gained from the load balancer. This was as important as the Busy Tone to our successful operations.

In spite of these limitations, the surviving contiguous chunk of relic data enabled us to not only reconstruct our previous exponential capacity forecasting model, but to improve on it with a more accurate power-law model constructed by virtue of gaining deeper insight while reviewing those data. Perhaps we should be grateful to Larry and Katrina[2] and take a congratulatory trip to Vegas. Nah!

References

[Bar03] A.-L. Barabási. *Linked: How Everything is Connected to Everything Else and What It Means for Business, Science and Everyday Life.* Plume, 2003.

[2] In case you (Dear reader) missed it, the abstract is a parody but the substance of this paper is not.

[CRT05] E. Candès, J. Romberg, and T. Tao. "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information". *IEEE Trans. Inform. Theory*, 2005. Submitted (`www.acm.caltech.edu/~emmanuel/papers/StableRecovery.pdf`).

[GNW80] N. J. Gunther, D. A. Nicole, and D. J. Wallace. "Goldstone modes in vacuum decay and first-order phase transitions". *Journal of Physics*, A(13):1755–1767, 1980.

[Gun01] N. J. Gunther. "Performance and scalability models for a hypergrowth e-Commerce Web site". In R. Dumke, C. Rautenstrauch, A. Schmietendorf, and A. Scholz, editors, *Performance Engineering: State of the Art and Current Trends*, volume # 2047, pages 267–282. Springer–Verlag, Heidelberg, 2001.

[Gun05a] N. J. Gunther. *Analyzing Computer System Performance Using Perl::PDQ*. Springer-Verlag, Heidelberg, Germany, 2005.

[Gun05b] N. J. Gunther. "Numerical investigations into physical power-law models of internet traffic using the renormalization group". In *Triennial Conference of the International Federation of Operations Research Societies*, Honolulu, Hawaii, July 11–15 2005.

[Gun06] N. J. Gunther. *Guerrilla Capacity Planning*. Springer-Verlag, Heidelberg, Germany, 2006. in press.

[HDA05] Q. He, C. Dovrolis, and M. H. Ammar. "On the predictability of large transfer TCP throughput". In *ACM SIGCOMM*, August 21–26 2005.

[Mal99] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, New York, NY, 1999.

[Yur98] Jack Yurkiewicz. Forecasting that fits. *ORMS Today*, February 1998.

APPENDICES

A ATO PROJECT DETAILS

The ATO project ran for about 12 months, costing about $2 million—half for software and half for systems and licenses. It was replaced by a $100,000 BEP developed system, before the ATO subsumed registrations into it's on-line environment. The original systems were decommissioned and the hardware transferred to an education site.

Initial registrations started around August 1999. The GST was introduced in Australia on 1 July 2000. ABN's were processed in a 2–4 week period. The ATO promised 2 week turnaround on electronic lodgment—either via the website or via the ATO's electronic links to Accountants. Paper based registrations were promised in 4 weeks. The dual-deadline caused confusion with the public shown in the main traffic peak of May 31, 2000 and smaller peaks at June 14 and June 30.

A.1 The Data

All performance data and capacity planning data were lost when the system was decommissioned. A CD containing backups of system "process accounting data" for the web servers and database backend for the period March 27 to September 13, 2000 was found.

The file format for the Sun Microsystems data was different to that on the systems available (Intel Linux and PPC Macintosh). The Sun format, a *struct*, was found documented in a $\mathcal{C}$ include file. Decoding this binary data had hurdles—the size of the record was two bytes longer than sum of the fields in the *struct*, and the binary values were stored in a different byte order (the classic Big-Endian/Little-Endian portability problem). Both $\mathcal{C}$ and Perl were used to create decoders, the final version used was in Perl.

Uncompressed and decoded, the raw data came to over 3 GB. The compressed raw files were 500 MB. Whilst data for the web servers was available, and we were interested in categorizing Web traffic, the problems of isolating registrations, meaningfully combining data from disparate systems and attempting to align the timestamps (clocks weren't synchronized) were sidestepped by looking at records from the database server as it provides a good proxy for whole system load.

A.2 The System

There were two Sun SMP web servers with a single Oracle database server on an 8-way Sun. An additional two slower and smaller web servers were included in the last few days. An Alteon content-switch/load balancer was used to manage the web servers and present a single IP number to the world.

As very sensitive data was being collected and handled, DSD (Defense Signals Directorate) who set Federal Government security standards, deemed dual firewalling was required - the web servers were firewalled from the Internet, and they accessed the database server through a second dedicated firewall. The administrative and backup systems along with a number of X-terminals were on the same LAN as web servers. System (tape) backups were done nightly by the Secure Hosting Facility staff in the secure area housing the servers, where they could log onto the systems via KVM connections to the graphical ports on the servers. Admin staff were allowed access to the terminals and admin machines in a less restricted area close by. The serial (RS-232) server consoles for booting were connected via a switch to a pair of serial terminals in the admin room.

The ATO mandated that the registration data could be transported to them by two means: e-mail as a primary and files on CD's as a secondary. Whilst this was made to work and tested, unfortunately the software contractor used different naming schemes for the two sets of files and did not provide the ATO with any way to reconcile them, nor rectified this when informed. E-mail delivery only was used throughout the life of the system. Each exported file contained data for just one ABN registration. Each email contained just one file and was acknowledged by a return email from the ATO when it had been securely stored by them for transfer to their backend processing systems. Resends were possible.

The application ran over SSL and was written as a CGI in a proprietary scripting language supplied by the software contractor. Significant systems, database and application performance analysis and tuning led to more than 20-fold improvement in system throughput from the originally delivered code. Oracle identified that the database server was consuming 50% of its CPU resource parsing the simple SQL queries. The software contractor would not modify their proprietary scripting language to address this.

The software contractors had 3 sets of servers for their development and testing environment. Their testing did not include any load tests or performance tuning.

Early traffic forecasts by the Operations and Admin team predicted 20,000 registrations on the final day. The only load estimate the ATO provided to the software contractors was "1,200 in a day", which they interpreted as 50 per hour, with no partial or abandoned applications, and no other work running on the systems.

When the ATO advertised the website on national TV over the last weekend in February, 2000 the system collapsed with a run queue of around 150 on the primary web server. A performance enhancement of the scripting language was brought forward to immediately address the issue. Additional CPU's and memory to double the capacity of the web servers were also ordered - with a 6 week delivery schedule.

As a permanent solution, the Admin staff designed a *Busy Tone* solution for the software contractor to implement. This was provided untested in the middle of May. Tuning the control parameters under live load conditions proved difficult, but was mastered within a few days. The software contractor ignored a key Busy Tone design requirement of "no database access".

A.3 Tools

A simple web-based performance reporting tool was constructed using RRDtool. System figures from `vmstat` and `iostat` were graphed. CGI script run times were extracted and reported as a proxy for webpage response time. The internal target was a 5 second response time. Other information such as number of transactions received and count of those sent via email to the ATO was extracted from the database and displayed as well.

Raw daily registration count data was downloaded and analyzed in Excel. Most of the time we were fixated on registrations completed per day, not measuring full system load.

The original load forecast data was taken from about 6 weeks of "low baseline" activity. The Christmas and early January holidays were not modeled, the traffic in this period was very much lower than the simple projection. Having the projection anywhere near the mark, let alone within 5%, seemed like blind luck.

Without the original spreadsheet, recreating the criteria and assumptions in the the initial projection hasn't been possible. It only makes sense to model on weekly data, as weekend traffic was significantly lower but the model produced an estimate for the highest daily load. A doubling period of around four weeks was found in the initial traffic.

A.4 Rate Limiting Factors

Early in the project, it was assumed that the Web servers would be the Rate Limiters and they were closely monitored. Oracle was engaged to monitor and tune the database server and the number of CPU's were doubled.

On the day of the spike, the Rate Limiter was a surprise - the second firewall, a 330 MHz Sun workstation with no performance monitoring enabled, was saturated. Secondary effects, such as the DNS timing out, were seen.

The best decision made was an unanticipated benefit as well—purchasing an Alteon Content-Switch/Load Balancer. This immediately scaled the performance of the front-end web servers, but more importantly, gave us transparent manageability of the system. It could have given the software contractors trivial and transparent upgrade/change process with true pre-production testing then instant 9 AM change-overs with trivial fall-back. They declined to avail themselves of the facility, preferring full-day outages on Sundays with limited testing.

What had been the main web server for the site, hosting 30+ Government web sites, continued to operate and have application upgrades during the registrations period. There was significant additional web server capacity which we were unable to utilize due to a routing problem—it was on a different subnet from the load-balancer. The server had enough ethernet ports to additional connect to the other subnet, and it was allowed by the Security Review Panel, but the system software was unable to route traffic back out the same interface—the whole system had just one "default route".

A.5 The Traffic

The site, and the majority of businesses using it, were located on the Australian East Coast. There were small influences from traffic in other time zones. Weekly traffic had distinct patterns (time in EST) :

- Weekends were quiet, about 60% of a normal day.
- One day of the week would be busier - Tuesdays initially, then Thursday.
- There were discernible peaks and troughs in the daily load:
  - From 2AM to 6AM, overnight traffic was low or zero.
  - Traffic started building around 6AM. By 9AM the site was busy.
  - There were two daily peaks - morning and afternoon, separated by lunchtime. Small 'blips' were visible on these peaks, delayed by the West Coast time difference.
  - Traffic declined steadily through the afternoon, reaching a low around 6PM, where it started building again to the evening peak around 8-9 PM.
  - Traffic fell away rapidly after midnight.
- Three unexpected, though obvious, effects were seen:
  - There was no Friday evening peak - people socialize.
  - There was no Saturday morning peak - it's a time for non-business activity.
  - The Sunday evening peak was as high or higher than a normal weekday - small businesses prepare for the coming week. This tended to end later than normal - 10 or 11PM.

Prior to the available data, the following distinct traffic periods were observed:

- Early: August 1999 to mid-December 1999. Low activity, steadily increasing with a 3–4 week doubling.
- Xmas Break: Mid-Dec 1999 to end-Jan 2000. Very quiet and chaotic - almost no traffic some days.
- Y2K. The web site was closed for registrations on Friday 31-Dec 1999.
- Pre-Advertising: 1-Feb 2000 to Sun 27-Feb 2000. After the holidays, traffic quickly picked up to where it had left off. Doubling period again 3-4 weeks.
- Site Overload: Mon 28-Feb and Tue 29-Feb 2000. Not a spike, caused by response to advertising swamping the site. Unexpected "cascade mode" failure.

Several distinct behavioral regions can be identified in the available traffic data:

- Post-Advertising: Wed 01-Mar to Tue 30-May 2000. Steadily increasing traffic. Traffic for the week including Easter shows a marked impact, with the volume, not peak, made up shortly after.
- Busy Tone active: 15-May to 31-May 2000. Exceeded 5 second response time. Users asked to retry.
- Spike: Wed 31-May 2000. Very high load in response to perceived deadline.
- June 2000: 01-June to 30-June 2000. Decreasing load, with a small spike about 16-June, the real deadline
- Post Deadline: 01-Jul 2000 to end of data in September 2000. Traffic steadily declining.

B GLOSSARY OF TERMS

The following terminology is used throughout this paper.

ABN: Australian Business Numbers registered with the ABR.

ABR: Australian Business Register.

ATO: Australian Taxation Office.

BEP: Business Entry Point.

Connection: Database connection. Each CGI script connects once to the Oracle database.

Email: Completed registrations are emailed to the ATO for processing. There is one email per message.

Message: Data for each registration sent as a single file, typically as one file per email or one message per file on a CD. Email was the default transport mechanism.

Page: Web page served by the CGI application.

Registration: Completed ABN registration filed with the ATO by business.

Transaction: Includes partial and abandoned attempts at registration.

Troppo: Diminutive form of *tropical*. Australian slang for *tropical* Australia. To have reached a state of tropical madness. The expression *gone troppo* probably derives from heading (north) into the vastness of tropical Australia and losing all sense of proportion due to the heat.